Thesis/
Reports
Glassy,
J.M.

Maya System Guide

Version 1.11

by Jospeh M. Glassy

FINAL REPORT FOR RESEARCH JOINT VENTURE AGMT
#INT-89380-RJVA with
Systems for Environmental Management
FS Contact: L. J. Lyon
Co-op Contact: Joseph M. Glassy

MAYA System Guide

Version 1.11

September 1, 1991

Project INT-89380-RJVA

Joseph M. Glassy
Systems for Environmental Management
PO Box 8868
Missoula, Montana 59807
(406) 549-7478

Table of Contents

Page

1 Introduction 1
- Philosphy and Goals 2
- Audience 2
- System Organization 2
- Hardware Requirements 3

2 Basic MAYA Concepts 3
- Overlay Types Supported 3
- Map Coding Conventions 4
- Map Attribute Tables 6

3 Understanding MAYA's Data Dictionary 6
- What Is A Database Table? 6
- The Data Dictionary: MAYA's Heart 7
- The MAYA Triumvirate: An Inseparable Trio 7

4 Spatial Analyses Supported in MAYA 11
- Point In Polygon Identification 11
- Point To Edge Distance Measures 12
- Sample Frame Analysis 13
- Roving Windows Analysis 14

5 Using MAYA: Overview Guide To Operation 16
- Installing MAYA 16
- Starting MAYA 16
- A Typical Sequence of Operations 17

6 The MAYA Front-End Reference Guide 20
- Data Menu Option 20
- Edit-View Option 21
- GIS Services Option 21
- Analysis Services Option 22
- Reports Menu Option 24
- System Menu Option 24
- Quit Option 26

7 Ad-Hoc Tools 26
- PC ARC/INFO 26
- FoxBASE Database Manager 27
- EPPL7 Raster GIS 27
- Documentation for Ad-Hoc Software Components 28

Appendix A: Data and Problem Limits 29
Appendix B: MAYA Data and Process Flow Diagram 30
Appendix C: Raster Map Formats Supported in MAYA 31
Appendix D: Tutorial Map Set Description (Maps 1-4) 32
Appendix E: A Brief MAYA Acronym and Term Glossary (use MIPS glossary) 33
Appendix F: Further Reading: An Annotated GIS and Image Processing Bibliography 34
Appendix G: Database Structures for the MAYA Triumvirate 35
Appendix H: An Optimal Hardware Configuration 39
Appendix I: MAYA.FOX Menu Tree Diagram 41

Acknowledgements

The MAYA software is the product of several years of cooperative research between the U.S.D.A. Forest Service, Intermountain Research Station in Missoula, Montana, and Systems for Environmental Management, a private, non-profit research group also based in Missoula, Montana (Research Joint Venture Agreement INT-89380-RJVA, SEM-92). L. Jack Lyon of the Intermountain Station was responsible for the basic concepts, motivation, and funding that made this implementation of MAYA possible. The author wishes to thank him for this vision, and for the dedication and uncommon patience shown throughout this project.

Etymology

For the curious reader, the term MAYA is not an acronym; it is also not used in the Central American Indian context of the word. MAYA is used here in the Hindu context, wherein the word might be loosely translated as "the illusion of the world's permanence".

1 Introduction

MAYA is a collection of software tools for evaluating and analyzing wildlife habitat. The software is designed to run on commonly available "PC" microcomputers based on the MS-DOS operating system. As a system, MAYA consists of four subsystem elements:

- a *front end* interface, MAYA.FOX, through which the user performs most spatial analysis functions. This interface is a FoxBASE program, and thus requires either a run-time version of FoxBASE or the full Developers version. MAYA is usually shipped with the run-time FoxBASE module, providing users with a minimal execution environment; serious user's are encouraged to independently purchase the complete FoxBASE database product.

- a spatial analysis *engine*, MAYASAMP.EXE, written in the ANSI C language, that performs all the low level operations. This engine has a command line interface, and is usually run from within the MAYA front end; most users do not directly interact with the engine.

- a small collection of support utility programs, used for converting various map formats to and from one another, and for other assorted spatial data manipulation tasks not handled directly by the MAYA spatial analysis engine.

- an ad-hoc component consisting of a collection of commercial support tools, which might include PC ARC/INFO GIS by ESRI, ERDAS image processing software by ERDAS Inc., the EPPL-7 raster GIS, PMAP, and/or other commercial GIS programs.

At the heart of the MAYA concept is a flexible microcomputer database manager, which is used to organize, retrieve, and arbitrate all data flows going into and out of the various MAYA subsystems. The database aspect of MAYA is discussed in some detail later in this document. MAYA does not require any commercial products to run, per se; users with raster data sets generated by supported commercial may find the core MAYA subsystems (front-end, engine, and utilities) to be a sufficient set of tools to accomplish a variety of spatial analysis tasks. But again, our recommendation to serious users is to consider purchasing at least one more general purpose GIS and/or image processing software product.

MAYA offers the wildlife researcher a set of tools that provide two different views into research problems: an ad-hoc view, and a more structured view through the use of specialized applications programs. Specifically, four types of point-on-polygon raster overlay analyses are offered. These two methods of investigating research problems ideally work in a complementary manner; assuming that research objectives and an initial hypothesis exist, the ad-hoc perspective can provide an unstructured environment in which to later frame the problems investigated through the more specialized applications.

1.1 Philosophy and Goals

MAYA was designed with the following general goals and assumptions in mind:

o	Usable By Non-Experts	Wildlife researchers should not have to be computer experts or programmers to enjoy the benefits of modern wildlife analysis technology.
o	Transparency	Software tools should operate as transparently as possible in the work environment, presenting the user with a robust, consistent, and unobtrusive interface between the research data and the tools used to manipulate the data.
o	Highly Integrated	Useful analysis software should attempt to tightly integrate a wide variety of tools in as seamless a manner as possible, so that users can move from one related task to another with as smoothly as possible.
o	Scaled Support	The software should attempt to scale its user support (e.g. context help, and menu driven access to functions, etc.) to match the needs of an evolving user, rather than adopt a "one size fits all" interface strategy.

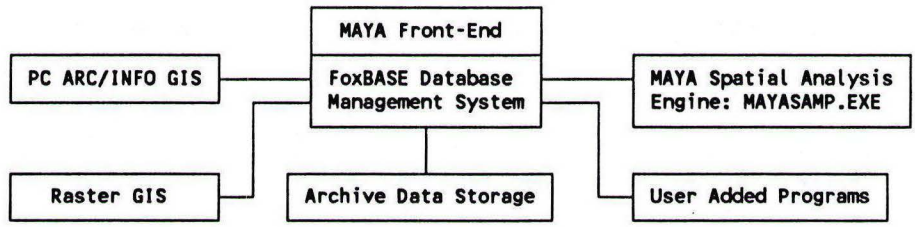
1.2 Audience

MAYA is designed for use by wildlife research professionals. Prior experience with computers is helpful but should not be essential. MAYA provides a work environment that attempts to promote incremental learning, whereby simple tasks and operations may be largely carried out by the novice through the use of the menu system. Later, when the user gains additional experience and insights, more complicated tasks may be tackled using a combination of the menus, ad-hoc mode, and user-defined batch jobs.

1.3 System Organization

The schematic shown below illustrates how the various MAYA subsystems generally relate to each other. In the ad-hoc mode many other cross-connections between the tools are possible.

Figure 1: System Schematic



Most of the MAYA's tools may be used in either an structured or ad-hoc manner. In the structured context, the front-end program (MAYA.FOX) shields the user from many of the intricacies of the subordinate tools. The menu driven "outer shell" serves as an intelligent control center from which almost all other MAYA facilities can be reached. A critical task that the MAYA front-end provides for users is to help manage the large number of separate files required to perform even a simple analysis.

The front-end contains several features to help streamline repetitive core tasks, such as those relating to file management. For example, MS-DOS limits filenames to a total of 12 characters (8 for the first segment, with an optional 3 character extension; remembering these often cryptic names can pose a serious (and we might argue, un-necessary) challenge to novice and experienced users alike. MAYA provides a facility to allow users to enter long (up to 48 character) descriptions for all file objects maintained in the main data dictionary. Once this is done, users may then select these files using scrollable, pop-up "pick-lists" showing the longer form of the file's name. Another, more subtle feature of the MAYA front-end is that most menu options are displayed in a "natural" order of operation, where such an order exists, so that the user is encouraged to perform sequence-dependent tasks in an efficient order.

Using MAYA's tools in an ad-hoc context assumes a greater familiarity and experience with the intricacies of the various programs. The main rationale behind the ad-hoc mode is that experienced users interested in directly performing a given task should not necessarily be forced to traverse a series of menus when they know what they want to do, and how to do it, in a more direct manner; thus, MAYA allows these users access the subsystem level tools directly.

1.4 Hardware Configuration Requirements

The MAYA software runs on IBM PC/AT class microcomputers. This means that the microcomputer must run the MS-DOS operating system, version 3.3 or higher, or a close variant of it. At the very minimum, MAYA requires an "AT" class (e.g. 80286 CPU @ 12mhz) microcomputer with 640K of RAM, with at least one serial port (for a mouse or digitizer), and one Centronics protocol parallel port (for an attached line printer or laser printer). The computer should have at least one floppy drive (either a 5.25" 1.2MG high density floppy or a 3.5" 1.4MG high density microfloppy), and a hard disk of no less than 40MG. Although MAYA will run on a system equipped with only a monochrome video system (e.g. Hercules at minimum), it is highly recommended that at least an EGA class video system be employed; a VGA graphics adaptor is highly encouraged for both clearer text displays and graphics displays.

Ideally, the computer system should possess the same peripherals required to efficiently run PC ARC/INFO. These include access to a line printer (or laser printer), connection to a digitizer tablet capable of running in point mode, and connection to a flat bed size C,D or E pen plotter. Users interested in our notion of a more "optimal" hardware configuration are encouraged to consult Appendix H. at the end of this document.

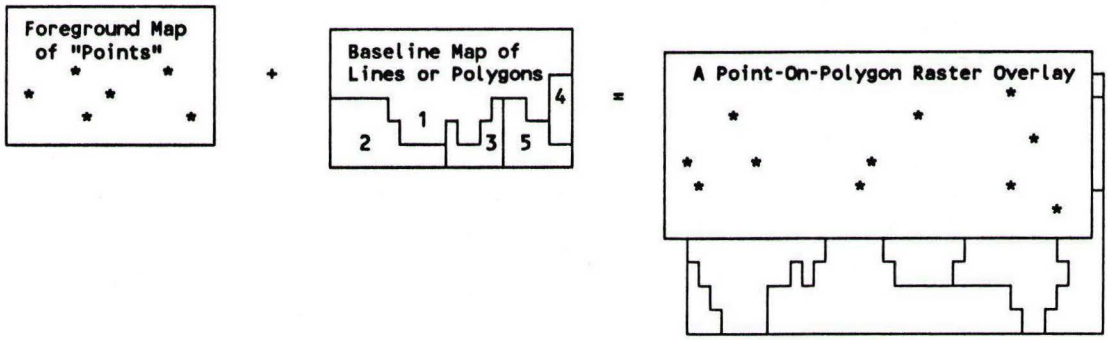
2 Basic MAYA Concepts

2.1 Overlay Types Supported

MAYA is designed to perform a narrowly defined, specific set of spatial analysis operations; it therefore should not be viewed as a generalized spatial analysis tool, even though within the confines of what it can do there is considerable flexibility in setting up the problems. All of the MAYA spatial analysis functions assume that the user will perform a simple, two layer raster overlay, where is "point" raster map is to be

overlain a "polygon" (or "linear") raster overlay. This will be generally referred to hereafter as a "point-on-polygon" overlay. Figure 2 below represents a symbolic illustration of such an overlay:

Figure 2: A Point-On-Polygon Overlay



2.2 Map Coding Conventions

Within the context of wildlife habitat research, the point, or foreground, raster map will often represent a series of animal locations, observed and mapped directly in the field or captured through radio-telemetry. The polygon, or baseline, map typically represents the habitat property of interest, often a classification of the habitat by vegetation type or some other meaningful ecosystem measure. Note that baseline maps may optionally be linear maps, representing roads classified by a density or use measure, or streams, classified by stream order or any other appropriate hydrologic measure.

Within MAYA, raster maps are assumed to be made up of square shaped grid cells (e.g. 30 meters on a side) forming a rectangular matrix. Each raster cell contains exactly one value, referred to by various GIS and image processing software alternately as the *cell value*, the *parity* value, the *DN* (data number), the *feature ID*, or the *cell attribute*. For our purposes, we will almost always refer to raster cells as referencing a *feature ID* value. There is a critical distinction between map features, and the attributes ascribed to these features; users who grasp this fundamental distinction find themselves a long ways towards mastering basic spatial analysis concepts. The two terms are defined and contrasted below:

Map Features — The term "map feature" refers here to the basic spatial units that the map represents. We define exactly three types of map features: point objects, line objects, and polygon objects. Each map feature is a unique, (that is non-repeating), entity, and is therefore given a unique identification number. Each map feature also by definition possesses a unique spatial position, relative to all other map features. It is the notion of a map features position, in terms of its relative and absolute spatial adjacency and contiguity to other features, that underlies the usefulness of all spatial analyses.

Feature Attributes — The term "feature attribute" refers to a property of a map feature. The term *attribute* is very often confused with *feature*, since a raster map cell may be coded to either type of value. A simple way to remember the difference between these two is that *attribute values* refer to the actual information content (category class or some measurement

variable) of a cell owned by a map feature, and that a *map feature* simply refers to the unique "container" of that information.

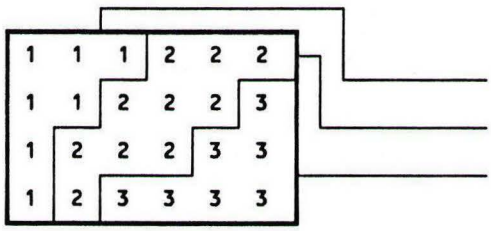
In the traditional GIS lexicon, map layers each contain only one type of feature (point, line, or polygon), the boundaries of which are defined in terms of common membership in one of the mapped classification categories. Either type of map, vector or raster, may be coded by feature ID; the key concept to remember about most GIS maps (vector or raster) is that their constituent units are most often defined in terms of a discrete set of categories, such as vegetation type categories, road density use classes, percent canopy cover, etc.

On the other hand, unclassified remote sensing data products, most often represent continuously varying values across a spatial extent. For example, LANDSAT Thematic Mapper (TM) image scenes have cells whose DN (cell values) represent some measure of the spectral properties of the region viewed; these are most definitely not map features, but should instead be considered a type of "attribute" (or property of) each grid cell. In an unclassified (raw) TM image, one will often find a continuously varying series of DN values in direct proximity to each other; while one might be interested in obtaining the frequency distribution of these spectral values, attempting to use these as though they represented discrete *features* would be a mistake.

In more sophisticated spatial analyses, map layers may also represent either discrete or continuously varying mathematical quantities, representing cost (or friction) surfaces, ease-of-travel measures, or use density measures. In this context, map layers may be used as individual terms in multi-layer "map algebra" expressions that result in other map layers. MAYA does not currently support such multi-layer map algebra.

At the simplest level, we may consider raster maps as representing two broad categories -- classified cells vs. unclassified cells. By convention, unclassified cells are coded 0, and denote regions either unmapped or outside the area of interest. Classified cells are conventionally given some positive integral value. In MAYA, raster map grid cell values usually serve as a simple key (or index) to the *unique map feature* the cell belongs to; the cell's value itself typically carries no real "map" information whatsoever. Thus, linear and polygon raster maps in MAYA usually contain contiguous regions of "same valued" cells, for each distinct feature. To retrieve the actual information associated with a given cell, the system uses the feature ID value as a search key into an external attribute table, where has one database record is defined for each map feature. The following Figure 3 illustrates this concept:

Figure 3:
Raster Map With 3 Features Linked To Attribute Table



Feature ID	Attribute	Other fields..
1	10233	aaaa
2	11432	bbbb
3	11834	cccc

In the above Figure 3, all the cells in feature 1 lie adjacent to each other, as do the cells in features 2 and 3. The connecting lines from the map feature(s) to the corresponding record(s) in the map's attribute table

illustrate how the feature ID (1,2 or 3) is used as a search key into the attribute table, pointing to where the map's real "information" is found, contained in the Attribute field (values 10233, 11432, and 11834).

2.3 Map Attribute Tables

While this indirect coding method may appear cumbersome at first, its flexibility allows map features to possess a variable number of attributes or properties, whose information value may exceed the maximum magnitude allowed to store in a single raster cell. The attribute value pointed to by each feature ID can also be a character string (up to 8 characters long) or a floating point (decimal) value.

The number of fields (minus one) in the map's external attribute table defines the upper limit of separate attributes that may be associated with the feature. Note that it is possible in MAYA to define baseline maps whose grid cell values are direct *attribute* values, thus eliminating the need for an external attribute table.

In MAYA, map attribute tables are standard FoxBASE database tables that contain at least two fields, a feature ID field (default name FEAT_ID), and an attribute field (default name ATTR). These table are usually ordered by feature ID field, in ascending order. When a given raster map is introduced to MAYA, if it has an associated attribute table, this table is also introduced at the same time. Specific details on how maps and attribute tables are "introduced" to MAYA is provided later in this document.

3 Understanding MAYA's Data Dictionary

Wildlife habitat research, like most types of research, is intrinsically a data intensive activity. Despite the difficult tradeoffs in choosing one type of database manager over another, the basic need for some type of well designed DBMS seems to remain constant regardless of the research task's complexity. Wildlife research data is simultaneously concerned with both the spatial and temporal state of its subjects, and current GIS technology uniquely presents an analysis environment pliant to both of these needs. GIS software typically reference data in both a spatial (map) context and a tabular (list) context. The formats of these two types of data (one in a binary arc-node or raster cell, and the other in database tables) are currently different enough, however, to prevent easily maintaining them in a single database management environment. This may change in the future with the advent of graphically oriented databases. Today, however, the challenge for GIS software is to make this physically disjunct relationship appear to the user as logically integrated as possible.

3.1 What is a Database Table?

Database tables may be simply considered matrices of information, organized into rows and columns. The rows are referred to as *records* and the columns are called *fields*. The familiar dBASE III+/IV (and FoxBASE, Clipper, dBXL etc.) file format is used for all of MAYA's database tables. This means that any of these other commercial products (as well as many other utilities) may be used to access information in any of MAYA's tables. All DBMS tables are maintained in *binary* form, which means that they cannot be interrogated using a standard text editor, or the DOS TYPE command, like ASCII text files can.

Database tables in MAYA are used for two basic purposes; to maintain internal system data (e.g. system tables) and to maintain user defined map and attribute data (data tables). System tables contain information

primarily for internal use by MAYA.FOX only. Users rarely, if ever, need to directly manipulate the data in system tables. Data tables, on the other hand, contain information obtained from the user on the various objects MAYA manages.

PC ARC/INFO Version 3.2.1, on the other hand, uses its own database manager (DBMS), a product by Henco called INFO. MAYA was not designed to use INFO as its database manager due to many serious limitations in the INFO software. MAYA does offer some limited utilities to translate its (.DBF) tables to and from Henco INFO tables. An upcoming release of PC ARC/INFO (version 3.4D) will formally end all ties to the older INFO database; at this point, all PC ARC/INFO database tables will be in the standard dBASE III+/FoxBASE format.

In addition, FoxBASE DBMS database tables in MAYA are used to store many other types of information within MAYA, including:

- Attribute data for Vector and raster maps
- System configuration and batch job facility information
- Names of external user-assigned DOS programs, executable from within MAYA
- Analysis results ready to be viewed and/or formatted into final reports

3.2 The Data Dictionary - MAYA's Heart

At MAYA's heart is a single table (MAYADD1.DBF) that contains a summary of all critical database tables and map objects managed at any one time within the system. This table is referred to as the table level data dictionary, because it stores all relevant characteristics for all system and data tables used in MAYA and maps; it even stores information on itself! Such information includes each table's DOS name, user provided label, MAYA table type, current storage format (native DOS or compressed), location on the DOS disk media, and date-time it was most recently updated.

All maps, PC ARC/INFO vector coverages, and attribute tables, and analysis result tables must appear in this data dictionary for MAYA to "know" about them, even if some or all of these reside on the DOS disk in a subdirectory. Most tables and maps are "introduced" to the data dictionary automatically by menu driven functions in MAYA, so in most cases, the user does not have to explicitly "enter" the information for a map or table directly into the dictionary. If the data dictionary record in MAYADD1.DBF associated with a particular critical table is deleted by accident, MAYA is "blinded" to the existence of this table, and operations requiring the table will fail. A separate discussion is included later in this documentation on maintaining the integrity of the data dictionary.

3.3 The MAYA Triumvirate: An Inseparable Trio

Two other system tables work with the main table level data dictionary table to provide MAYA's baseline analysis services. Together, these three tables (the MAYA Triumvirate) manage and organize a catalog of all maps and attribute tables required for any given analyses. These tables are all related to each other via the ID code (integers) assigned to each file in the data dictionary automatically by MAYA. They are extremely interdependent, like Siamese triplets. Each of the three tables listed below is briefly introduced here, and discussed in more detail in the User's Guide.

Triumvirate Table	Purpose
MAYADD1.DBF	Main table level data dictionary
MAYARDEF.DBF	Raster map characteristics table
MAYAPARM.DBF	Analysis Input Parameter sets

3.3.1 MAYADD1.DBF - The Table Level Data Dictionary

MAYA needs to know certain facts about each map, attribute table, and result table managed by the system. Such facts include the files name, DOS subdirectory location (full path, to allow to smooth functioning across multi-disk volume computers), type, storage format, status, and most importantly of all, the unique (integer) ID number assigned by the system to the file. By storing such operating-system dependent information in one central location, the need for redundant storage in other tables is greatly reduced. It also provides the user with a single convenient place to look for a catalog of all files known to the system at any one point in time.

Each record in the MAYADD1.DBF contains exactly one record for each critical table required by the MAYA system. Tables and/or files considered "critical" include vector coverage (names), raster map files (either foreground or background maps, stored in either .DBF format or binary format), foreground and background map attribute tables, and analysis result tables. Note that no map or table ever appears twice in this dictionary; if it does appear more than once, this is an error and will require correction!

3.3.2 MAYARDEF.DBF - Raster Map Definitions

Each raster map possesses certain characteristics relevant to the entire map. These include the maps unique ID and type code (e.g. point, line or polygon), the resolution of raster cells (e.g. 45 meter square), the number of raster rows and columns making up the map, the unique ID of the vector map (e.g. the parent vector map that spawned the child raster), and the geographic coordinates denoting the raster map outermost bounds, expressed as a rectangle, usually in UTM coordinates.

Each record in MAYARDEF.DBF contains the collected definition for exactly one raster map. The unique ID for the raster map is used as a relational key into the MAYADD1.DBF table, significantly reducing the storage of redundant information in MAYARDEF.DBF for each map.

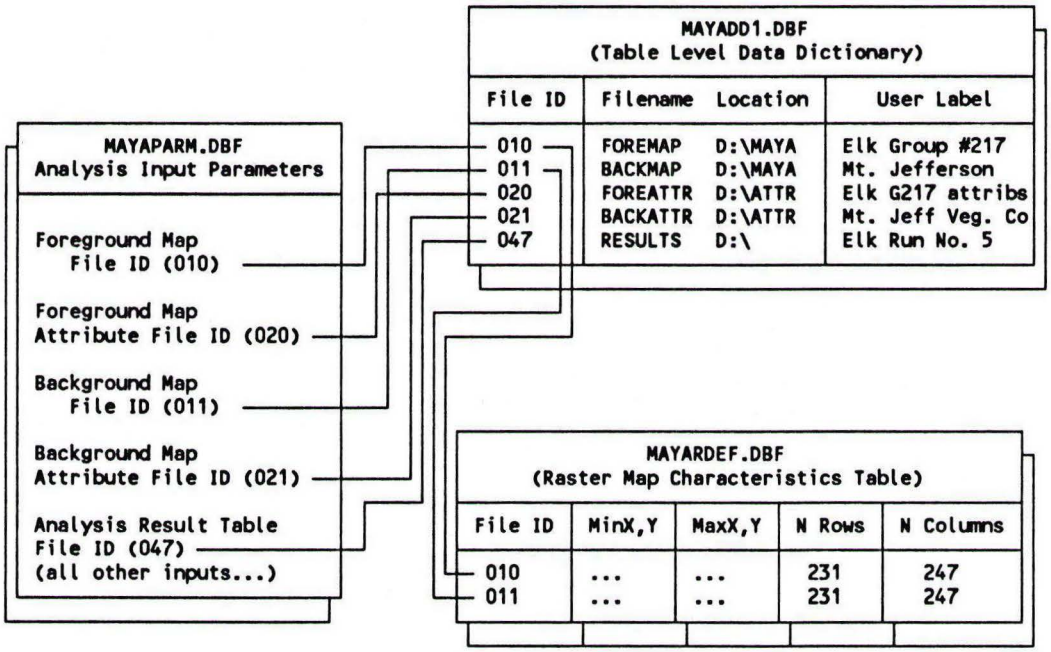
3.3.3 MAYAPARM.DBF - Analysis Input Parameters

Each analysis requires input parameters to guide the execution of the run. These include the type of analysis, the IDs of both the foreground and background raster maps, the IDs of the foreground and background map attribute tables, and the settings for various switches governing the runtime behavior of the analysis.

Each record in MAYAPARM.DBF contains the input parameters for exactly one MAYA spatial analysis. Users enter the parameters using a full-screen form within the MAYA front-end. The analysis inputs are stored in a DBMS table this way for two reasons: a) to take advantage of the superior data entry environment provided by FoxBASE, allowing range and type checking of all inputs at entry-time, and b) to facilitate batch execution of many analyses, run back-to-back.

The role that each of the triumvirate tables play is described in detail in the technical documentation. Below is a symbolic diagram illustrating their relationship to each other; note that not all fields actually present in the tables are shown below!

Figure 4: Dictionary Relationships



The above diagram illustrates the critical nature of the FILE_ID code assigned to each map (and attribute table) managed by MAYA. This FILE_ID functions as the relational "key" (read glue) that binds these three tables together in a dynamic way. The use of these relational "keys" allows the system to minimize the redundant storage of information required to manage each file, such as the file's name, its location on the hard disk, a user assigned text description, and other state fields.

In summary, for MAYA to work correctly, each map (and attribute table) is assigned a unique FILE_ID automatically by the system when it is created, which is then used in a variety of relational contexts. Note that the user of MAYA need not ever be concerned with knowing the FILE_ID of any particular file; indeed, it is the system's job to insulate the user from such mundane details! Instead, the user selects map(s), attribute file(s), and/or result tables using longer text descriptions from a scrollable pick list.

Note also that the table level data dictionary, MAYADD1.DBF, contains information on all tables required later by the stand alone analysis engine (MAYASAMP.EXE), and that the raster map characteristics table, MAYARDEF.DBF, contains information only relevant to raster maps. Whenever a MAYA analysis is performed, the following sequence of events takes place:

(1) The input parameters for each analysis run are stored as one record in the MAYAPARM.DBF table. When an analysis is performed, the user selects the analysis by name from a pick list, and the MAYA front-end converts this to a record number in MAYAPARM.DBF where these inputs are stored, and supplies this record number in MAYAPARM.DBF to the program about to execute.

Storing sets of input parameters as multiple instances in a table this way also easily allows for batch execution of any number of runs, since a driver program can issue a range of records to the analysis engine(s) to execute.

(2) The analysis program next looks into the table level data dictionary (MAYADD1.DBF) for the necessary information on the foreground and background maps and attribute tables, each of which is referenced by its FILE_ID code. The table dictionary provides the program with the filenames and DOS path (e.g. location on the hard disk) so that it can open the files for processing.

(3) Rather than store each raster maps "header" information in the map data file itself, this "map-wise" information is stored, one record per map, in the raster map characteristics table. The analysis program uses the maps' FILE_IDs to find and lookup this data from the MAYARDEF.DBF table. The consolidation of all raster map header data, such as map size, cell resolution, etc, into one table also makes it easy for user's to quickly gain an overall perspective on all raster maps in the system.

3.3.4 Triumvirate Summary: Integrity Issues

The MAYA triumvirate (MAYADD1.DBF, MAYAPARM.DBF, and MAYARDEF.DBF) forms an interdependent trio of tables, none of which can really get along without the other. This is good and bad; it some respects by dividing up critical data this way into three files, it serves to better protect the integrity of the data in each (instead of storing it all in one massive file, for example). The downside of this approach is that it does place a great deal more burden of maintaining the system integrity on the correctness of the FILE_ID values; should these become corrupted, the whose house of cards could fall. Additional measures for ensuring the inter-table synchronicity are being explored, and will be implemented in future releases of MAYA if possible.

4 Spatial Analysis Procedures Supported In MAYA

MAYA currently provides four types of raster based spatial analyses procedures. Each of these is briefly introduced below:

4.1 Point In Polygon (PIP) Identification

The Point-In-Polygon Identification analysis (hereafter abbreviated as PIP) simply returns the attribute description code (polygon or line classification) of the baseline map layer cell underlying a set of foreground layer cells, which often represent animal locations. The classification code returned typically represents a habitat type, or vegetation community type code) indicating the nature of the habitat occupied by the animal at that point in time. Alternatively, the attribute (code) returned may serve as a key into a separate database table that contains a series of attribute fields. These fields could represent all the data items collected at a fixed sample plot, or any other collection of database fields in the record referenced by the key code. Note that the attribute code returned is stored as a character (string field in the attribute table, allowing it to represent either an integer, or an alphanumeric code.

Most of the "point identification" analyses seek to answer set intersection questions of the form "...given a point on the foreground layer, what corresponding point on the baseline layer shares a common geographic

location, within the tolerances of a single raster cell?" Sample objectives for the various combinations of PIP analyses are provided below:

Point on Point	"...how many of the foreground map points directly overlay (on a cell basis) a corresponding point on the baseline map layer."
Point on Line	"...which point(s) on the foreground map directly overlay a classified line feature on the baseline map. Or, what are the characteristics of line features on the baseline map that are overlaid by points on the foreground map."
Point On Polygon	"...for each point in the foreground layer, what are the characteristics of the underlying polygon feature it occupies on the baseline map."

4.1.1 How the Point-In-Polygon Analysis Works

Each foreground map point, (1..N_FEATURES), is successively retrieved. The attribute code corresponding to each foreground feature is looked up from the external attribute table, and the raster address for the point is used to retrieve the spatially correspondent cell in the opposing (baseline) map layer. Note that the baseline map layer may be any feature type (point, line, polygon) but the foreground map layer must be a point map.

The attribute for the spatially correspondent baseline map cell, underlying the current foreground cell, is retrieved, and a result database record is written out, containing the foreground and baseline map cells feature ID's and attribute codes.

4.2 Point-To-Edge Distance Analysis

The point-to-edge distance analysis is a type of proximity search. It produces the euclidian distance (e.g. as the crow flies) from each successive point feature in the foreground layer to the nearest feature edge on the baseline map layer. The baseline map feature edge located may be a polygon edge, an intersection with a linear feature such as a road or stream, or may even be a point feature on the baseline map.

Additionally, a setwise criteria option is provided, which allows the proximity search to calculate the distance from each foreground point to the first member of a user-specified set of baseline map feature attributes encountered. This allows users to define a set of "functionally equivalent" attribute types, any of which can terminate the search.

This analysis generates a database table that contains the foreground cell feature ID and attribute, the baseline map cell's feature ID and attribute that was closest to the foreground cell, and the (minimum) distance between the two real map units (usually meters). Unlike many common proximity search functions in GIS software such as PMAP, ERDAS, EPPL7, IDRISI this search function does not stop at some pre-defined maximum search distance; instead, it will continue to probe until one of the map edges is encountered.

Analyses involving point to edge distances answer questions relating to how far a given point is a point, line, or polygon edge, on the map in the opposing layer. The most common type of question relates to how close,

rather than how far, a foreground point might be from the (nearest) point or feature edge in the baseline map. Here are some possible analysis objectives:

Point on Point	"...for each point on the foreground map layer, what is the distance, in map units, to the nearest point on the baseline map layer", or, "..for each point on the foreground map layer, what is the distance from this point to the nearest point on the baseline map layer belonging to any one of the categories that forms the (user defined) acceptance set of attributes?"
Point On Line	"...for each point on the foreground map layer, what is the distance, in map units, to the nearest linear feature on the baseline map layer", or, "..for each point on the foreground map layer, what is the distance from this point to the nearest linear feature on the baseline map layer belonging to any one of the categories that forms the (user defined) acceptance set of attributes?"
Point On Polygon	"...for each point on the foreground map layer, what is the distance, in map units, to the nearest polygon edge on the baseline map layer", or, "..for each point on the foreground map layer, what is the distance from this point to the nearest polygon edge on the baseline map layer whose polygon belongs to any one of the categories that forming the (user defined) acceptance set of attributes?"

4.2.1 How the Point-to-Edge (PTE) Distance Analysis Works

The PTE procedure cycles through a set of foreground cell locations and a set of qualifying criteria, and find the nearest qualifying (e.g. "different") cell ("edge") relative to each successive foreground map cell. This is done by an "inward-out" spatial proximity search, originating at the foreground cell and successively radiating outward, in a spiral pattern, until the second closest qualifying baseline cell is encountered. The search terminates as soon as the minimum cartesian distance of the first two matches found; the minimum distance is by definition the smaller of these two distances.

In order to qualify as "sufficiently different" from a given foreground cell, a baseline map cell must not be an unclassified cell, and it must be a member of the "acceptance set" associated with the foreground map layer. Initially, all baseline feature attribute codes are members of the acceptance set, allowing the search to terminate as soon as the first qualifying (e.g. nearest) baseline map cell is encountered. Users may optionally define the acceptance set as a subset of the whole set, rendering all attributes but the chosen members functionally "invisible" for the purposes of the search.

When the search terminates, the nearest opposing map layer feature edge has been located, and a euclidian line is "snapped" and measured between the identified foreground and baseline map cells. The length of this line is the minimum distance desired, and is written out, in map units, as a record in the resulting database table; output records also include the cell information at either end of the minimum distance line -- the foreground and baseline map cells' feature ID and attribute codes. At the end of the procedure, exactly one result table record has been produced for each discrete point map feature in the foreground layer.

4.3 Sample Frame Analysis

The sample frame analysis offered by MAYA is a spatial sampling procedure. It allows the user to project an elliptical or rectangular "sample frame" about a set of foreground map layer cells (points). This basic operation is similar to the symmetric "feature buffering" overlay capability offered by many commercial GIS software packages; the main difference is that the sample frame are generated "on the fly" (not requiring a separate coverage layer for each such frame) and may be any symmetric geometric shape.

Analyses involving sample frames seek to provide estimates of areal frequency, within the sample frame, of one or more baseline map features. In MAYA, the foreground map is used in a variety of ways to define the position and limits of the sample frame. A *sample frame* is best envisioned as an imaginary polygon feature (e.g. a spatial mask) applied over an opposing map layer, which may represent point, line, or polygon features. The sample frame serves to identify an area within which the analyst needs to know the areal frequency of features found on the opposing (baseline map) layer.

Sample frames in MAYA are symmetric geometric figures, which may be rectangular or elliptic in shape. If elliptic, they may have their major axes oriented towards any compass position. The centroid of each such sample frame is defined by the position of the foreground map feature (point). In the case where a polygon map is used as a foreground layer, the sample frame is not a symmetric, but rather has its boundary defined as the irregular perimeter of each foreground polygon itself. Examples of possible objectives for each combination of map feature are provided below:

Point On Point	"...produce a frequency distribution of the area occupied by all baseline map point features occurring within the bounds of the sample frame, whose centroid is defined as each foreground map point. Obtain the absolute, relative, and cumulative frequencies, in map unit areas, found within the frame".
Point On Line	"...produce a frequency distribution of area occupied by all baseline map linear features (e.g. roads, streams, etc) occurring within the sample frames.
Point On Polygon	"...produce a frequency distribution of area occupied by all baseline map polygon features occurring within the sample frame, whose placement is determined by each point in the foreground map layer.

4.3.1 How the Sample Frame Analysis (SFA) Works

The interior and edge cells of each such sample frame are sequentially scanned, resulting in a frequency distribution of baseline map layer attributes encountered within the frame. The class frequencies for each discrete baseline attribute encountered are written as result database records, with one record per baseline attribute encountered. These tables are ordered by the foreground cell feature IDs used as frame centroids. The frequency statistics returned include the absolute, relative, and cumulative frequencies of baseline feature attributes, expressed in square raster map units.

4.4 Roving Window Frequency Analysis

The roving window (frequency) analysis is similar to the sample frame analysis described above, except that the projected sample frames are centered about synthetic points determined by the program itself, according

to a user specified sampling parameterization. The roving window routine offers two basic methods for dispersing the set of points forming the centers of the sample frames; a) systematic strip sample method, and b) uniform random point placement.

4.4.1 How the Roving Windows Analysis Works

The systematic strip sample method places a regular matrix of frame centroids beginning at the north-west most corner of the map, moving from west to east across and down the remainder of the map. The result, when square shaped sample frames are employed, is a "super" gridding of the map, producing an exhaustive sample of the map. Note that if rectangular frames are used, the sample is close to a 100% spatial sample; if elliptic shaped frames are used, the intersticial regions are not sampled, and thus the sample does not represent a 100% spatial sample. The size, shape, and orientation of the sample frames is determined by user specified parameters found in the analysis inputs database table.

The random method results in a user specified density (or absolute number) of randomly placed points being generated, representing centroids for a uniform pseudo-random spatial sample. The sample frames, whose positions are determined by the randomly dispersed points, behave identically to the sample frame analysis described earlier.

5 Using MAYA: Overview Guide To Operations

5.1 Installing MAYA

Installing MAYA involves several simple steps: a) preparing the computing environment to run MAYA, and b) copying all MAYA distribution software and tutorial data to the destination computer. MAYA requires one core subdirectory (usually called \MAYA, off the root file system). We make the assumption here that the correct version of FoxBASE has previously been installed, and that its directory has been added to the system PATH. Each of the steps required to install MAYA is discussed in further detail next.

5.1.1 Preparing the computing environment for MAYA

New MAYA users need to add a DOS environment variable, create a core subdirectory named MAYA, and adjust their system PATH to include the MAYA core subdirectory, to optimally run MAYA. The install program described below only provisionally makes these adjustments to the operating systems "semi-permanent" state; to make the changes more "permanent" users need to add these changes to their AUTOEXEC.BAT file (the file that controls the standard start-up environment for the operating system).

(1) A DOS environment variable named MAYA needs to be defined, and set equal to the unambiguous DOS path where the MAYA core subdirectory is located. To define MAYA subdirectory on drive D:, one would enter the following at the DOS command prompt, as well as inserting an identical line in the AUTOEXEC.BAT file:

```
SET MAYA=D:\MAYA
```

(2) The MAYA core subdirectory should be created, matching the name entered in the above step 1:

```
D>MKDIR \MAYA
```

(3) Last, the AUTOEXEC.BAT file should be edited. The PATH statement should have the following added to the end of it: as in: PATH=. . . . ;D:\MAYA

5.1.2 Installing the MAYA software

Once the computing environment has been prepared as described above, the MAYA distribution set of files can be copied to the destination computer. To do this, insert the MAYA distribution disk in drive A: and enter MAYAINST <destination-drive>. A batch file will copy all files to the indicated drive.

5.2 Starting MAYA

To start MAYA, just enter MAYA at the DOS command line. For example, if logged into drive D:, the user would type:

```
C>MAYA <carriage return>
```

Note that this will work only if the DOS PATH has been edited to include the MAYA core subdirectory, and if the DOS environment variable, MAYA, has also been correctly defined as described above. When begun, the MAYA front-end program will be executed by the FoxBASE database program. If the MAYA core subdirectory is not placed in the globally visible DOS PATH environment variable, the user must explicitly log into the MAYA core subdirectory prior to invoking the program.

5.3 Using MAYA: A Typical Sequence of Operations

MAYA is a flexible spatial analysis environment that may be used in a number of ways. There is, however, a general sequence of operations that is typically followed. This sequence is described here, providing a starting point for new users. It is not meant as a rigid "cookbook", because different users may find themselves entering and exiting the following process flow at different points.

The MAYA interactive front-end is specifically designed to make these routine operations efficient and reliable; the intrepid user who wishes to may also enter the values collectively defining an analysis directly into the appropriate MAYA database tables. Most users are encouraged to use the front-end program, MAYA.FOX, to perform these duties. Refer to Appendix B for data and process flow diagram that visually illustrates some of these relationships.

5.3.1 Basic Steps To Using MAYA: One Suggested Sequence

- o Identify Analysis Objectives
- o Establish Map Database
- o Assemble map attribute data
- o Introduce map and attribute metadata to MAYA
- o Define analysis parameters
- o Execute the analysis
- o View the analysis result table
- o Incorporate MAYA analysis results in larger spatial analysis

Each of these steps is described in more detail next.

5.3.1.1 Identify Analysis Objectives

Most users will want to start by defining the overall analysis objectives; it should go without saying that this step highly influences all subsequent steps! Unfortunately, many analyses are unduly driven by data availability, and other esoteric constraints (e.g. hardware availability) rather than by theoretical aspects of the problem. Availability of good quality spatial data is, has been, and will probably continue to be a real problem for spatial analysts; recognizing the impact that such challenges pose, better studies will result when the overarching questions drive the methodology.

5.3.1.2 Establish map database

Once the basic objectives are established, the map database specific to the analysis may be assembled. These may originate as GIS products from an external spatial processing environment (such as ESRI's ARC/INFO, PMAP, IDRISI, EPPL7, etc). If the map data is not already in a digital form, which is most often the case,

the map must first be digitized, usually into some type of vector (arc-node) format. Any maps in a vector format must then be rasterized before bringing into MAYA, into one of the basic raster formats supported by MAYA. Refer to Appendix C for a description of raster maps supported in MAYA.

5.3.1.3 Attribute Data Assembly

Associated with each maps is usually a list of feature attributes, comprising the "legend" to the map. These must be assembled in dBASE III+/FoxBASE database tables, subject to the constraints imposed by MAYA (see Appendix A for *Data and Problem Size Limits* imposed by this version of MAYA). A separate attribute database table is usually defined for each distinct map used within MAYA. For data products originating in some commercial GIS environments (e.g. ERSI's PC ARC/INFO) the Point or Polygon Attribute Table (.PAT) provides an excellent starting point. Other packages, such as EPPL7, also have a similar table or file.

At a minimum, there should be one record for each map feature (point, line, or polygon classified). There should be at least two database table fields defined, one for the feature ID (by default named FEAT_ID, unless the user overrides with a different name), and one for the feature's attribute code (by default named ATTR, unless overridden). The feature ID field is defined as a Numeric field, 5 characters wide. Use a code of 0 for unclassified map regions if possible, and positive integer values for classified features. The attribute code is an 8 character field that may contain a number or any sequence of alphanumeric characters.

5.3.1.4 Introducing Map and Attribute Table Metadata To MAYA

The format and physical characteristics of each map to be used must be "introduced" to MAYA, to allow the system to efficiently track and manage the map data within the analysis environment. Maps are introduced to MAYA within the front-end program, MAYA.FOX, using a full screen entry form, which stores the map data in the raster map characteristics table, MAYARDEF.DBF.

Attribute tables built previously must be explicitly "introduced" to the MAYA data dictionary before any actual analyses may be run. The name and location of all attribute tables is entered in the MAYA table level data dictionary (MAYADD1.DBF), within the front-end program. Refer to the MAYA Menu Reference Guide later in this document for information on the specific menu options required to introduce map and attribute data to the data dictionary.

5.3.1.5 Define Analysis Parameters

Using the maps and attribute tables introduced above, various analyses may now be defined. The front-end program, MAYA.FOX, contains a full screen entry form to enter analysis parameter data. Each analysis is given a record in the MAYAPARM.DBF table. Analysis input parameters include the type of analysis, all run time options governing how the analysis is to be run, and identifiers for all map and attribute tables required to perform the analysis. Refer to Appendix G for a description of the database structures for the MAYA triumvirate tables: MAYADD1.DBF, MAYARDEF.DBF, and MAYAPARM.DBF.

5.3.1.6 Execute MAYA Analysis

After all maps and attribute data is introduced to MAYA, and at least one analysis is defined, an analysis may be executed. The front end program calls the external MAYASAMP.EXE program, which writes an appropriate result database table. Appendix G documents the formats of all three types of result database tables; note that the sample frame and roving windows analyses produce identical format result tables.

5.3.1.7 View Analysis Results

After an analysis has been run, the user may wish to interactively view the result table created. This is done within the MAYA.FOX front-end program, or may be done off-line in any program that supports browsing dBASE III+ tables. A standard set of pre-defined "report formats" are available within MAYA to produce text file or hardcopy prints of any of the result tables produced by an analysis.

5.3.1.8 Incorporate Results

Analysis results in database tables are then usually incorporated in any external analysis, as appropriate. Since MAYA is oriented towards spatial sampling, the results may be used as end products themselves if the project is merely a descriptive study. For studies involving hypothesis testing and further statistical analysis, the tables MAYA produces are usually considered intermediate data.

6 *A MAYA Front-End Reference Guide*

The interactive, menu drive front-end to MAYA, MAYA.FOX, presents the user with a hierarchical menu structure. Horizontal bar menu items are selected via the familiar "bounce-bar" menu methods popularized by spreadsheet programs and other commercial PC software: the arrow cursor keys may be used to position the highlight to a particular choice, or the items first letter may be chosen. A diagram of the complete MAYA.FOX menu tree may also be found in Appendix I, providing a useful "quick-reference" to old and new users alike.

The main (horizontal) menu consists of seven choices, each leading to a vertical pop up menu that groups similar activities:

Data **Edit-View** **GIS Services** **Analysis Services** **Reports** **System** **Quit**

Descriptions of the specific menu options are found below, organized by these seven major headings. A globally accessible, limited context sensitive help system is also available via the F1 HELP hot key.

6.1 Data Menu Option

The Data main branch is used to perform most ancillary data management tasks associated with common MAYA tasks. Under the Data option are the following six submenu choices:

DOS Directory List	This option generates a pop-up window containing a user specified wildcard for a DOS directory listing. Any wildcard may be specified, and may optionally include a different disk drive than the currently logged one.
Convert Data Services	This option provides some limited data conversion services, allowing MAYA DBF and Henco INFO tables to be converted to/from each other. It also provides a way to convert raster files to a DBF format.
List Coverages	This option generates a list of PC ARC/INFO vector coverages available, at the currently logged in subdirectory level.
Archive Services	This option opens up another pop-up menu, to allow for Importing/Exporting PC ARC/INFO (.E00) file coverages, compressing or de-compressing a set of user specified files, or invoking a user defined multi-diskette backup program.
Telecommunications	This option invokes a user defined telecommunications software program, to allow the user to transfer files to/from a remote host computer.
Maintain Data Tables	This option opens up another sub-menu, providing these services: viewing DBF file structures, appending ASCII data to existing DBF tables, conditionally replacing data table field values, conditionally deleting data records, modifying data table structures, and erasing DBF, text, or INFO files.

6.2 Edit-View Menu Option

The Edit-View main branch is used to passively view or actively edit either ASCII text files or dBASE III+/FoxBASE database tables. In addition, new text files and/or new database tables may be created in under this menu. The user is first prompted for the type of file to operate on (text or DBF), and then is prompted for the type of operation to perform, from these choices: Select, View, Edit, Create. When Select is chosen, a pop-up "pick-list" of files of the appropriate type is displayed; a new wildcard may be specified anytime.

Text files are passively viewed using a user defined BROWSE utility (such as Borlands README utility, the public domain LIST utility, or any other simple text file browser). Text files are actively edited with a user defined (and supplied) text editor, such as VEDIT, Brief, Sage SPE, PCWRITE, etc.

6.3 GIS Services Menu Option

The GIS Services main branch offers a limited interface to the PC ARC/INFO command set, and provides gateways to critical GIS related tasks in MAYA. The following submenu choices are offered:

PC ARC/INFO Command Access

This option generates a pop up list of PC ARC/INFO commands. The default is to show all commands, although users can easily trim this set down to a "commonly used" list, since the list is stored in a database table. Commands that require arguments can accept them via prompts to the user. Upon selection, the chosen command is executed; after it is finished the user is returned to the MAYA front-end.

Describe Coverage

This option prompts the user to enter the name of a PC ARC/INFO coverage to obtain summary information for. The standard DESCRIBE command is run, with output optionally directed to a text file.

Rasterization Services

This option opens up another sub-menu, which provides a gateway to introducing new raster maps to the MAYA system, viewing the characteristics of existing raster maps, deleting raster maps, or invoking the PC ARC/INFO rasterizing functions. The submenu contains these choices:

- Select an Existing Raster Map Definition
- Create New Raster Map Definition
- Edit Selected Raster Map Definition
- Delete Selected Raster Map Definition
- Begin Rasterization Now

Generate Coverage

This option prompts the user for the name of an ASCII text file containing vector data to be used to generate a new PC ARC/INFO vector coverage. The GENERATE command is then invoked, building the new coverage.

Quick Plot View	This option is used to obtain a "quick" view of an PC ARC/INFO vector coverage in graphics mode, using the EDITPLOT command.

6.4 Analysis Services Menu Option

The Analysis Services main branch provides a gateway to all MAYA analysis related tasks. Four submenus are displayed at this option, each of which is described in detail next. If a new analysis definition is to be created, one of the four choices is selected:

- Identify Point in Polygon
- Point To Edge Distance Analysis
- Sample Frame Analysis
- Roving Windows Analysis

Once at least one analysis definition is present, taken from the choices above, the startup services menu is used to fill in the entry.

6.4.1 MAYA Raster Analysis Functions

The Startup Services option includes four sub-choices, allowing the user to choose an existing set to work with, view-edit the selected set, delete the selected set, or execute the analysis indicated by the selected analysis set.

6.4.1.1 Choose An Analysis Input Parameter Set

This option presents a pop-up list of currently defined analysis definitions, read from the MAYAPARM.DBF table. User provided analysis labels are used to identify the analysis input set to work with.

6.4.1.2 View-Edit Selected Analysis Input Parameter Set

This option displays a full screen data entry form, for entering all required parameters for a given analysis. The prompts are context sensitive to the type of analysis requested.

6.4.1.3 Delete Selected Analysis Input Parameter Set

This option allows the selected analysis input set to be deleted from the data dictionary. Once done, the definition cannot be un-deleted.

6.4.1.4 Run Selected Analysis NOW

This option invokes the selected analysis. The MAYASAMP engine is called, and a status screen is displayed while the analysis is running, periodically reporting the relative progress through the task. After the sample analysis is done, control returns to the MAYA front end program.

6.4.2 Analysis Batch Job Services

This option opens up another submenu for batch analysis processing, presenting the following choices. Batch processing is useful when a number of related analyses have been defined, such as for a sensitivity analysis, and all analyses may be executed in a group.

6.4.2.1 Enable/Disable Analysis Input Sets

Selecting this choice presents a browse table view to allow any logical subset of currently defined analysis input definitions to be toggled ON or OFF for the purposes of executing (or deleting) the definitions. A "Y" is entered to toggle a particular analysis ON and a "N" (or blank) is entered to toggle an analysis OFF.

6.4.2.2 Delete All Disabled Analysis Input Sets

Selecting this choice causes *all* analysis input sets not currently toggled ON (or TRUE) to be deleted from the MAYA data dictionary. Use this option with care, as once it is done they cannot be easily brought back. We suggest that users produce a written report of the analysis definitions to be deleted first, before deleting them.

6.4.2.3 Run ALL Enabled Analysis Input Sets

Selecting this choice causes all analysis input definition currently toggled ON (or TRUE) to be executed, one after the other. This is an active command, and causes execution to begin immediately.

6.4.3 DOS Batch Job Services

This menu option opens up a submenu, for creating, selecting, or deleting multi-line DOS batch jobs. Any DOS executable task may be named in these jobs, and they may be as many lines as the task requires. Each line is a separate DOS task.

6.4.4 Run A User Assigned Program

This menu option provides a way to "shell-out" to any DOS program previously entered into the MAYAPROG database table through the MAYA front-end program. A pop-up list containing each user defined program appears when this option is selected. When an external program is selected, MAYA is to a temporary disk file, yielding all but about 5K of RAM to the child DOS process to run; when the child program is done, control returns to MAYA.

6.5 Reports Menu Option

The Reports main branch is used to obtain unformatted and formatted (dBASE/FoxBASE .FRM style reports) reports. The resulting report is sent by default to the LPT1: line printer port. The report menu includes these choices:

Database file print	This option prompts the user for the name of the database table to generate the report from, and the name of the particular report format (e.g. .FRM file) that contains the report specification.
Text file print	This option prompts the user for the name of the text file to generate the report from; either an unformatted (raw), or formatted report may be produced. If a formatted report is requested, the file is filtered through a user defined text file report utility.
Most Recent Report	This option allows the user to quickly interactively browse through the most recently produced result table, from a MAYA analysis.
Begin Printing Report	This option sends the report specified by the above parameters to the attached line printer.

6.6 System Menu Option

The System main branch provides a gateway to a set of MAYA system related services, including setting or inspecting system defaults, viewing analysis log entries from past jobs, and accessing the data dictionary tables directly.

6.6.1 Log System Access

Selecting this option opens up a BROWSE session for the log table, MAYALOG.DBF, that contains text of system generated log messages produced before and after each MAYA analysis is run. Starting, stop, and elapsed times are included, along with a label describing the task performed.

6.6.2 Data Dictionary Services

Selecting this option opens up the Data Dictionary submenu, which provides a gateway to direct access of the MAYA data dictionary tables. Choices for this menu are:

6.6.2.1 Select Table or Field Dictionary To Use

This option prompts the user to select which dictionary table to access. The table level dictionary contains metadata on all file objects used within MAYA; the field level dictionary contains all database table field definitions, including their data type, width, names, etc.

6.6.2.2 View-Edit Dictionary Table

The View-Edit option opens up a BROWSE session for the selected dictionary table, allowing the user to view and/or directly edit entries in either the Field or Table level dictionary. Editing these tables should be done with caution, since many MAYA components expect data to be formatted in a fixed way.

6.6.2.3 Produce A Dictionary Report

This option is used to generate a report on the selected dictionary table. The report is output to the standard line printer at LPT1:, using a .FRM style format.

6.6.2.4 Restore Dictionary Integrity

This option is used to refresh all system database table indexes used throughout the system. It is provided in the event that one or more tables become "out-of-sync" with their indexes. The most common way this happens is if some change is made to a critical system table outside of MAYA; changes made within MAYA automatically post to the index files affected.

6.6.3 Edit User Program List

This option provides a high level entry point to the database table containing a list of user defined external (DOS executable) programs that the user may wish to invoke from within MAYA.

6.6.4 Configure Defaults

This option opens up a BROWSE session for the MAYACNFG.DBF table that contains all system wide defaults affecting how MAYA operates. Through this table, the user may specify his/her own text editor, backup program, telecommunications program, browse program, and a variety of other system defaults.

6.6.5 System Status

This option presents a summary listing of internal memory or database table usage within MAYA. It is normally not required except in debugging situations.

6.6.6 Run Any DOS Command

This option is used to enter any single-line, standard DOS command to run "one time only". These entries are not saved after execution. To save a frequently run DOS command, or sequence of commands, the DOS Batch facility should be used.

6.7 Quit Menu Option

The Quit main branch is used return to the DOS operating system after a given session in MAYA is complete.

7 Ad-Hoc Tools

One of the basic ideas behind developing MAYA was to provide wildlife researchers with a dynamic environment to perform commonly needed spatial analyses. Although MAYA does offer the (four) analyses types described above, it is unrealistic to suppose that any one system could be designed to include all the tasks helpful to researchers under one roof. For this reason, the MAYA system encourages the use of the core tools in an ad-hoc manner.

The researcher is provided with three basic kinds of tools which may be used in an ad-hoc as well as structured manner: a) the PC ARC/INFO GIS by ESRI, b) FoxBASE DBMS, and c) a generalized raster GIS environment. Originally, we attempted to interface MAYA directly to the PMAP raster GIS; upon encountering severe limitations on the size of raster maps that PMAP could handle, we made a late switch to the EPPL7 raster GIS; the current state of the MAYA-EPPL7 interface is minimal at best. If MAYA is developed further in the future, a high priority will be to extend the capabilities of the interface to EPPL7, or select another raster GIS where such an interface can be readily developed.

In addition to the "standard" tools, provisions exist in the system to integrate the user's choice of statistical analysis software, telecommunications software, spreadsheet software, word processing software, and any other in-house DOS applications. Each of the standard ad-hoc tools are broadly described below.

7.1 PC ARC/INFO GIS

There are few limitations in the way MAYA can be used for ad-hoc explorations of spatial wildlife habitat relationships. The ambitious user willing to rigorously investigate the PC ARC/INFO software and extensive documentation will find a rich set of interactive capabilities for creating, editing, and analyzing vector based two dimensional spatial objects, referred to within ARC/INFO as point, arc, and polygon coverages. The most recent "host" version of ARC/INFO (Version 6.0) also offers a comprehensive raster data modelling capability, called GRID. Although PC ARC/INFO (Version 3.4d) does not currently offer any raster based analysis capability, vector coverages in ARC/INFO can be easily rasterized, up to a maximum resolution of 5000 rows by 5000 columns, and brought into other environments (such as EPPL7) and/or DBMS tables for further manipulation and analysis.

The GIS modules supplied by ESRI most relevant to MAYA include:

- PC ARC/INFO Starter Kit
- Grid Conversion Module
- Overlay Module
- ARC/EDIT Module
- ARC/PLOT Module

At this time, the Network Module available with PC ARC/INFO is not employed within MAYA. The Network module has the important potential to serve as an analysis engine for questions regarding wildlife travel paths, habitat use patterns influenced by forest road networks, and related questions.

7.2 FoxBASE DBMS

The FoxBASE DBMS, (Version 2.0) program is used as the gateway to all MAYA system and data tables. Used interactively, this software includes a menu driven command mode as well as offering the standard "dot prompt" command mode. FoxBASE operates on native dBASE III+/dBASE IV/FoxBASE tables. Using FoxBASE, any MAYA database table can be interrogated, and resulting list style and/or formatted reports may be directed to the screen, printer, and/or a disk file. FoxBASE offers a superior database performance level, considering that all numeric data it maintains is stored in an ASCII representation rather than binary. Especially helpful to the new user is the FoxBASE help system, a context sensitive help system assessable at any time.

The FoxBASE system also includes an excellent program design/generator tool, a database program documenter, and a pseudo-compiler. FoxBASE includes facilities for importing a variety of ASCII format data files into its database table format, including delimited (by blank, comma, or other symbol), or system data format (where data fields occurs in regular fixed column positions). As a testimony to the power of the FoxBASE database programming language, the entire MAYA.FOX front-end was written in FoxBASE.

7.3 EPPL7 Raster GIS

EPPL7 stands for the Environmental Planning and Programming Language. It was developed by the Minnesota State Planning Agency, and is a grid cell based Geographic Information System. The EPPL7 software is distributed by the Minnesota Land Management Information Center. EPPL7 raster maps may range in size up to 32,000 rows by 32,000 columns, although in practice you will rarely use maps this large. One limitation of EPPL7 is that the most attributes a given map may contain is 256 -- a sufficient number for many but not all wildlife based projects.

EPPL7 consists of a number of stand alone programs driven through a simple menu shell. The programs are divided into 6 main categories: a) Analytical commands, b) Resizing Commands, c) Combinatorial Commands, d) Conversion Commands, e) Display commands, and f) Utility commands. In addition, a utility called DOTPLOT is provided to produce hard copy output from screen displays. EPPL7 contains many of the same type of raster analysis commands found in other common raster GIS packages such as PMAP, IDRISI, and SPANS. Not all the routines are as comprehensive as the implementations found in the other packages, but they do work with larger maps. EPPL7 requires the industry standard CGI graphics device interface drivers; since PC ARC/INFO also uses this type of graphics driver, the two packages are compatible in this regard.

Noticeably absent in release 1.2 of EPPL7 (the current release) is any well designed interface to common DBMS software such as dBASE III+/IV or FoxBASE. Release 2.0 of EPPL7 is expected to at least partially address this lack, although it is not known which commercial database format, if any, will be followed. Map and color legends for EPPL7 maps may be stored in ASCII format however, which means that dBASE III+ tables in MAYA can be easily converted for use in EPPL7.

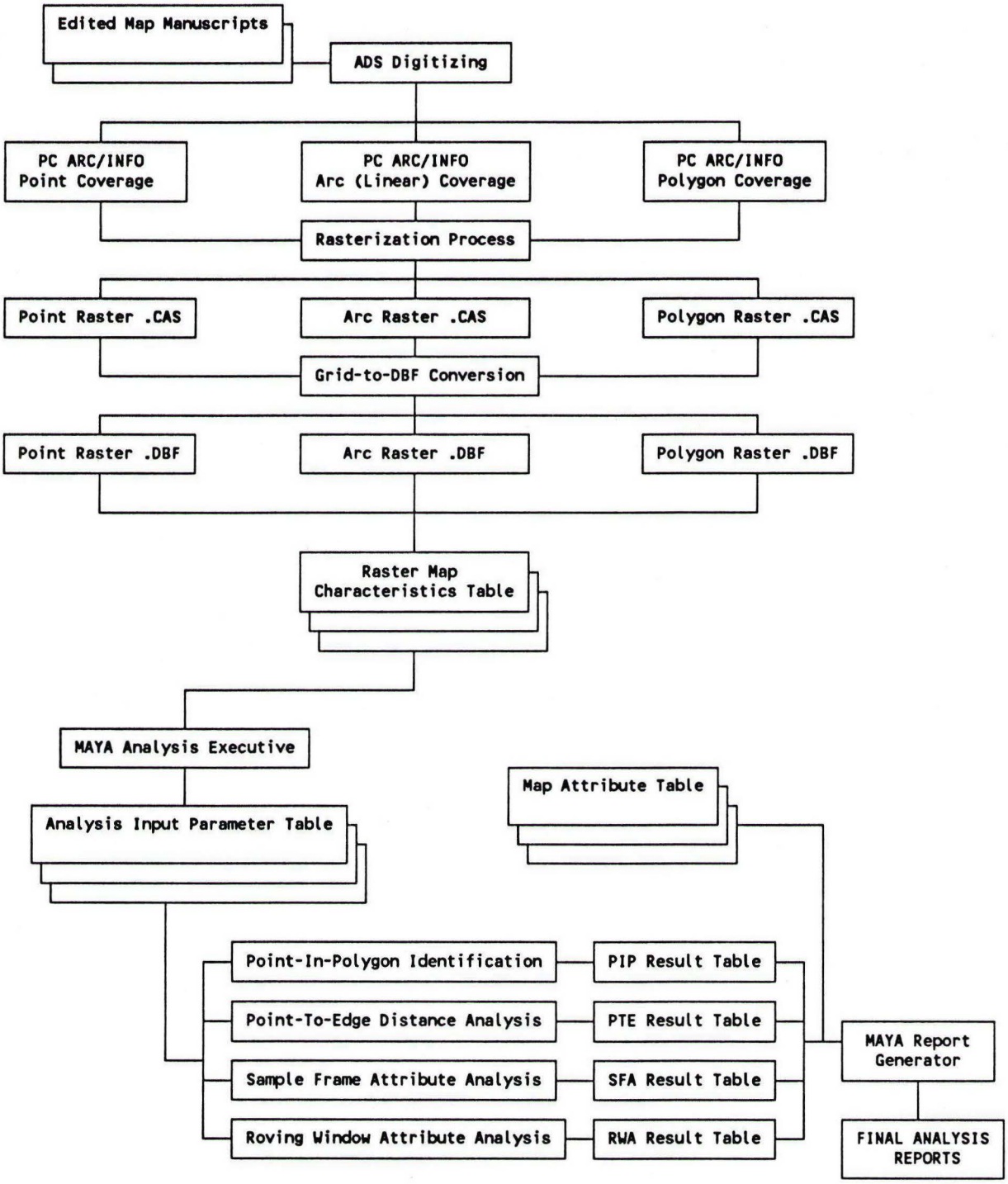
7.4 Documentation for Ad-Hoc Software Components

The documentation accompanying MAYA does not attempt to duplicate or comprehensively interpret the commercial documentation that comes with the above packages. Some of the basic functions that MAYA performs may be executed using commercial software in an ad-hoc manner, given sufficient time to decompose and reorganize the problem for the commercial system.

Appendix A: Data and Problem Size Limits

Maximum Number of Raster Map Rows	32767
Maximum Number of Raster Map Columns	32767
Maximum Number of Map Layers Simultaneously Overlaid	2
Maximum Number of Features Per Map	8192
Maximum Size of A Sample Frame	Size of Baseline Map
Maximum Number of Sample Frames	8192
Highest Allowable Feature ID	32767
Attribute Field Size Limit	8 characters

Appendix B: MAYA Data and Process Flow Diagram



Appendix C: Raster Map Formats Supported In MAYA

Format Code	Description
CAS	This is a ERSI PC ARC/INFO compressed (ASCII) raster format. Maps in this format must be converted using the MAYA utility called CAS2DBF.EXE before they can be used.
DRC_Bin	This is a "sparse" binary map format, primarily for storing "point" map layers; only classified cells are stored -- all unclassified cells are omitted. It is a standard dBASE III+/FoxBASE file format, including three fields: ROW, COL, and FEAT_ID.
DMO_Bin	This is a full orthogonal binary map format (non-sparse), where all cells, classified or not, are stored. Cells are stored in row-major order, meaning that all cells for all columns in row 1 are stored first, followed by all columns in row 2, and so on to the highest numbered map row.
RMO_Bin	This is a binary, orthogonal format where cells are stored in row-major order as binary 16 bit integer values. Each cells spatial position is implied by its row-major order in the raster.

Appendix D: Standard MAYA Tutorial Map Set

The tutorial map set serves two main purposes: it provides a "standard" set of test maps to use to gain familiarity with the system, and it provides a benchmark suite to verify that the system software has been properly installed.

Map File Name	Map Format	Rows	Columns	Features
TUTFORE1.DBF	DRC_BIN	20	20	5
TUTBACK1.DBF	DMO_BIN	20	20	4
TUTFORE2.DBF	DRC_BIN	60	60	20
TUTBACK2.DBF	DMO_BIN	60	60	9
TUTFORE3.DBF	DRC_BIN	100	100	90
TUTBACK3.DBF	DMO_BIN	100	100	31

Appendix E: A Brief MAYA Acronym and Term Glossary

Term	Definition
Feature Attribute	An attribute is simply a property of a map feature. It may be unique or non-unique to one or more map features.
GIS	Geographical Information System
Grid Cell	Smallest unit of data in a raster map; a square shaped geometric entity containing one value.
Map Feature	A unique map entity, usually representing either a point, line, or polygon object. Features have a topological context, meaning they exist somewhere in space. Features therefore have unique identifiers, so there may be only one of each feature on a given map.
PIP	Point-In-Polygon analyses
PTE	Point to edge distance analyses
Raster Map	A rectangular (matrix) of grid cells, usually stored in a binary, random access file, containing classified and unclassified cell values.
RCF	Row-Column format, a sparse, binary raster map format particular to MAYA that includes its own embedded spatial index. A RCF format data object contains a feature ID, a map row, and a map column for the cell.
ROV	Roving windows analyses
SCM	Single criteria (feature) masking. This term refers to the capability in MAYA to selectively mask out (temporarily make invisible) one or more features from either active map layer through a logical flag in the map's attribute table for the affected feature(s).
SFA	Sample Frame Analyses

Appendix F: Further Reading: A GIS/Image Processing Reading List

Burroughs, P.A. et al. 1986. Principles of Geographical Information Systems for Land Resources Assessment. Clarendon Press, Oxford, UK.

Campbell, R. 19xx. Introduction to Remote Sensing.

Issaks, Edward H. and R.M. Srivastava. 1989. An Introduction to Applied Geostatistics. Oxford University Press. New York, New York 10016.

Hammond, R. and P.S. McCullagh. 1986. Quantitative Techniques in Geography. Clarendon Press, Oxford UK.

Monmonier, Mark S. 1982. Computer Assisted Cartography: Principles and Prospects. Prentice Hall, Inc. Englewood Cliffs, NJ. 07632.

Starr, Jeffrey and J. Estes. 1990. Geographic Information Systems, An Introduction. Prentice Hall, Inc. Englewood Cliffs, NJ 07632.

Tomlin, C. Dana. 1990. Geographic Information Systems and Cartographic Modelling. Prentice Hall, Englewood Cliffs NJ 07632.

Appendix G: Database Structures for the MAYA Triumvirate

MAYA Triumvirate Database Table Structures

MAYADD1.DBF : the table level data dictionary

Field	Field Name	Type	Width	Dec	Description
1	FILE_ID	Number	5.	0	: Unique ID given to this file
2	FILE_NAME	Character	12.	0	: DOS filename and extension
3	USER_LABEL	Character	48.	0	: User provided descriptive label
4	UPDATED	Date	8.	0	: Date file last updated
5	UPDATETIME	Character	8.	0	: Time file last updated
6	DOS_PATH	Character	52.	0	: DOS subdirectory path for file
7	FILE_TYPE	Character	5.	0	: MAYA file type code
8	STATUS	Character	2.	0	: file's status in MAYA
9	ACCESSPRIV	Character	1.	0	: access priviledge code (not used)
** Total **			142		

MAYARDEF.DBF : the raster map characteristics table

Field	Field Name	Type	Width	Dec	Description
1	BATCH_TOGL	Character	1.	0	: Toggle (T/F) for batch processing
2	MAP_TYPE	Character	5.	0	: Raster map type
3	DEST_FORM	Character	3.	0	: Raster map format code
4	PARITY_FLD	Character	1.	0	: Cells coded by F,A(Feature,Attrib)
5	FEATFIELD	Character	10.	0	: Name Feature ID field in attrib.
6	ATTRFIELD	Character	10.	0	: Name of Attrib field in attrib
7	COVFILE_ID	Number	5.	0	: Vector coverage file ID
8	RASFILE_ID	Number	5.	0	: Raster maps file ID
9	BACKGRDVAL	Character	8.	0	: Unclassified cells value
10	N_ROWS	Number	5.	0	: No. of raster map rows
11	N_COLS	Number	5.	0	: No. of raster map columns
12	CELL_SIZE	Number	8.	2	: Cells width,height (in meters)
13	COVX_MIN	Number	10.	2	: Min X UTM location
14	COVY_MIN	Number	10.	2	: Min Y UTM location
15	COVX_MAX	Number	10.	2	: Max X UTM location
16	COVY_MAX	Number	10.	2	: Max Y UTM location
** Total **			107		

MAYAPARM.DBF : the analysis input parameters table

Field	Field Name	Type	Width	Dec	Description
1	TASK_LABEL	Character	48.	0	: User provided Analysis description
2	RUNTOGGLE	Character	1.	0	: Toggle (T/F) for batch processing
3	ANALYTYPE	Character	1.	0	: Analysis type code (P,E,S,R)
4	REPT_SCOPE	Character	1.	0	: Report scope (Detailed,Summary)
5	PTE_METHOD	Character	1.	0	: PTE analysis method code
6	SAMPLSHAPE	Character	1.	0	: Sample frame shape code (R,E)
7	ROVEMETHOD	Character	1.	0	: Roving windows method
8	SAMPLEMODE	Character	1.	0	: Sample frame method code
9	FOREMAPID	Number	5.	0	: Foreground map file ID
10	FORATTRID	Number	5.	0	: Foreground attrib table file ID
11	BACKMAPID	Number	5.	0	: Baseline map file ID
12	BAKATTRID	Number	5.	0	: Baseline attrib table file ID
13	RESULTID	Number	5.	0	: Result table file ID
14	USERTBLID	Number	5.	0	: user table ID (not used)
15	NUMSAMPLES	Number	5.	0	: Number of Roving Window samples
16	SAMDENSITY	Number	5.	1	: Roving window frame density
17	MAJORAXIS	Number	10.	2	: Sample frame major axis diameter
18	MINORAXIS	Number	10.	2	: Sample frame minor axis diameter
19	ORIENT_DEG	Number	5.	1	: Sample frame orientation (degrees)
** Total **			121		

Other Critical MAYA Database Table Structures

Structure for database : MAYACNFG.DBF (default values)

Field	Field Name	Type	Width	Dec	Description
1	USERNAME	Character	32.	0	: primary Users initials
2	TXTEDITOR	Character	48.	0	: users preferred text editor
3	BROWSEDIT	Character	48.	0	: users preferred browse utility
4	BACKUPPRG	Character	48.	0	: users preferred backup utility
5	ARKPACK	Character	48.	0	: users preferred compression util
6	ARKUNPACK	Character	48.	0	: users preferred decompression util
7	DFLTWILD	Character	32.	0	: default DOS wildcard for lists
8	DFLTBACKGR	Character	8.	0	: default map unclassified cell code
9	TEXTREPT	Character	32.	0	: users text report formatter
10	TELEPROG	Character	32.	0	: users telecommunication util
11	LOGSYSON	Character	1.	0	: Toggle (T/F) for log system
12	EXECPAUSE	Character	1.	0	: Toggle (T/F) to pause on execution
13	DEST_FORM	Character	3.	0	: default raster map format
14	CELL_SIZE	Number	5.	0	: default raster cell size
15	RASTER_GIS	Character	16.	0	: default raster ad-hoc GIS software
16	SHOW_DISKS	Character	1.	0	: Toggle T/F, showing avail disk,RAM
** Total **			404		

Structure for database : MAYADD2.DBF (field level dictionary)

```
Field  Field Name  Type       Width  Dec   Description
    1  FIELD_NAME  Character     10.   0 : Database field's alpha name
    2  FIELD_TYPE  Character      1.   0 : Database field's data type code
    3  FIELD_LEN   Number         3.   0 : Width of field
    4  FIELD_DEC   Number         3.   0 : No. decimals in field, if numeric
    5  FIELD_LABL  Character     36.   0 : Field's user provided text label
    6  FILE_TYPE   Character      5.   0 : MAYA file type code
    7  FIELD_POS   Number         3.   0 : Order of this field in table
** Total **                      62
```

Structure for database : MAYALOG.DBF (session log table)

```
Field  Field Name  Type       Width  Dec   Description
    1  SYS_MESG    Character     70.   0 : System generated log message
    2  STARTED     Character     17.   0 : Date and Time a task began
    3  FINISHED    Character     17.   0 : Date and Time a task ended
    4  ELAPSED     Character      9.   0 : Elapsed time HH:MM:SS task took
    5  MSG_CODE    Character      1.   0 : log message code
** Total **                     115
```

MAYA Result Table Structures

MAYAPIPR.DBF : Point in Polygon Identification Result table

```
Field  Field Name  Type       Width  Dec   Description
    1  PT_FEAT_ID  Number         4.   0 : Points feature ID
    2  PT_ATTR     Number         5.   0 : Points attribute code
    3  BK_FEAT_ID  Number         5.   0 : Baseline cells feature ID
    4  BK_ATTR     Number         5.   0 : Baseline cells attribute
** Total **                      20
```

MAYAPTER.DBF : Point to edge distance analysis result table

```
Field  Field Name  Type       Width  Dec   Description
    1  PT_FEAT_ID  Number         4.   0 : Points feature ID code
    2  PT_ATTR     Number         5.   0 : Points attribute code
    3  PT_BKATTR   Number         5.   0 : Baseline cell attribute under pt
    4  BK_FEAT_ID  Number         5.   0 : Baseline cell feature under pt
    5  BK_ATTR     Number         5.   0 : Baseline cell attribute (nearest)
    6  MIN_DIST    Number         9.   1 : Minimum pt-edge distance (meters)
** Total **                      34
```

MAYASFAR.DBF : Sample Frame and Roving Windows result table

Field	Field Name	Type	Width	Dec	Description
1	FRAME_ID	Number	5.	0	: Sample frame (point) ID
2	FORE_ATTR	Number	5.	0	: Foreground cells attribute code
3	CELL_ROW	Number	4.	0	: Foreground cells row position
4	CELL_COL	Number	4.	0	: Foreground cells column position
5	BACK_ATTR	Number	5.	0	: Baseline cells attribute
6	ABS_FREQ	Number	8.	0	: Absolute frequency of base attrib
7	REL_FREQ	Number	5.	1	: Relative frequency of base attrib
8	CUM_FREQ	Number	5.	1	: Cumulative frequency of base attr.
9	FRAMESIZE	Number	9.	1	: Frames size (not used)
** Total **			51		

Appendix H: An "Optimal" Hardware Configuration

Wildlife researchers who would like to obtain "better" performance from the MAYA software may be interested in our notion of an "optimally" configured hardware system. This may be helpful both to new sites involved in budget planning and hardware acquisition bids, and to existing GIS sites interested in upgrading their present equipment.

Like most software, MAYA performs basically two classes of computing operations: disk (or I/O) oriented, and compute (or CPU) oriented. Adding certain equipment to the computer can dramatically increase performance in both of these areas. Many GIS users would like to improve their equipments' performance. As critical as this might seem, it is even more important to understand the concept of *"balance"* in hardware resources. In short, "balance" means to understand how various components of the computer work together and variously constrain each others performance. The key to understanding "balance" is to grasp the concept of *throughput*. "Throughput" loosely refers to how much actual work gets done in a given period of time; it is a way of integrating the effects of disparate hardware components, each pushing and pulling at the limitations of each other towards the end result.

For example, it makes little sense to concentrate on improving strictly numeric processing capability if the real constraints on system throughput are due to relatively slow hard disk Input/Output (I/O). Conversely, it makes little sense to equip a computer with a relatively slow CPU with a fast hard disk; anytime one aspect of the computer hardware spends a great deal of time waiting for another component, an imbalance is present. Awareness of such imbalances helps users develop sensible, defensible upgrade priorities.

The next section involves some necessarily "technical" topics. Some of it may require some additional "interpretation" by a person more familiar with Intel based microcomputer hardware. The discussion is presented in terms of candidate hardware areas that could benefit the most from improvements.

CPU Type and Speed

Whenever possible, MAYA should be run on a computer equipped with one or the new Intel CPU chips such as the 80386SX, 80386 or 80486; these include advanced memory and task management capabilities somewhat beyond those found in 80286 based computers. Choices of clock speeds for these chips ranges from 16mhz to 33mhz (soon 40,50, and even 66mhz); in general, the faster the clock, the more system throughput. Higher speed CPUs will only benefit MAYA up to a point; as speeds go up beyond 25mhz, so do complications due to heat and slow memory chips, so make sure your machine's memory is rated compatibly with the CPU.

A secondary issue that dramatically affects overall system throughput is the choice of system bus. A bus may be simplistically viewed as the platform upon which the entire system is built; its characteristics and quality affect virtually every other aspect of the computers performance. The computer motherboard incorporates the bus circuitry at an integral level. Until somewhat recently, there were few choices in bus design available to consumers -- the IBM AT bus was the prevalent design, excluding the older S100 on its way out. With the advent of the IBM microchannel bus the alternative EISA bus, consumers now have greater choice than ever. Both of these bus designs offer a far superior platform relative to the AT bus. These newer bus designs offer increased signal bandwidth, faster system clocks, and intelligent arbitration of multiplexed bus signals. Until recently, there have been few add-on components available that took advantage of the characteristics of the new bus designs, but this is changing fast. In summary, if you can afford a computer with an MCA or EISA bus and are concerned with maximum performance, buy it.

Numeric Intensive Performance

GIS systems frequently perform graphics operations, and these almost always benefit from the presence of a numeric data coprocessor (e.g. NDP). In fact, you can't run PC ARC/INFO without an Intel or Intel compatible NDP installed. Until recently, Intel NDPs were the only choice; now other vendors have developed pin-compatible NDP chips which may be substituted for the 80287 or 80387, offering in some cases dramatically increased numeric performance (Cyrix, ITT). A key thing to remember is that NDP chips primarily boost floating point arithmetic operations (particularly transcendental functions such as sine, cosine, exponentiation,etc.) and offer virtually no improvements in integer arithmetic functions. Note also that if you buy a computer with a 80486DX CPU installed, it by definition contains a built-in NDP, so you won't need to add an external 80x87 NDP; 80486SX chips on the other hand do not contain built in math chips.

The speed of these chips should be roughly matched to the speed of the CPU, less 5%-10%. For example, there is some doubt that a 25mhz 80386 can effectively use all the capacity of a 25mhz NDP, so one rated at 20mhz would be adequate, with usually significant cost savings. Similarly, a 20mhz CPU can use a 16 mhz NDP.

Hard Disk Throughput and EMS Memory

Microcomputers generally spend a great deal of time waiting for I/O peripherals, and the hard disk is often one of the worst offenders. Therefore, efforts spent on improving this aspect of the system will deliver immediately noticeable performance improvement. The first line of defense here is to use a hard disk with good "native" speed. Disk speed is often summarized in terms of average track to track access time (in ms); a good access time is 14-16ms. Lower average access times roughly equate to better overall disk performance. In reality, performance will often be a function of how various packets of data are accessed (e.g. whether they are sequentially or randomly accessed, etc.), and on the type of disk controller used. The best hard disk controllers use an Enhanced Small Device Interface (ESDI) or Small Computer System Interface (SCSI) rather than the slower ST-506 MFM interface.

The easiest and least expensive way to improve disk performance (other than buying a fast disk) is to provide it with some type of disk cache. Usually these will be implemented as system software (loaded at boot time). Disk caches often use a type of least-recently-used algorithm, which means that the larger the cache (up to a point), the better the hit rate and thus, the better the overall performance. Since lower DOS memory (below 640K) is such a valuable commodity, it is recommended that MAYA user's equip their machines with as much EMS (Lotus-Intel-Microsoft (LIM) 4.0 EMS compatible) memory as possible. A total of 2 to 8MG of EMS is not unreasonable. The disk cache itself should be set up with at least 1MG of EMS.

Video Subsystem

Beyond the absolute minimum required (EGA), the quality of the GIS user's video subsystem is more one of personal taste. Most GIS related visual operations in PC ARC/INFO can benefit from a high resolution video subsystem. Digitizing is one task in particular that noticeably benefits from increased graphics resolution, for two reasons: a) the screen can display more of the map at one time, requiring less zooming, and b) the images shown can be displayed at much greater detail. The higher resolution adaptors also often tend to be faster.

The Enhanced Graphics Adaptor (EGA), at 640 by 350 by 16 colors is an absolute minimum; a much better configuration is to use a Video Graphics Array (VGA) board at 640 by 480 by 16 (or 64 or 256) colors. Super VGA boards commonly available now offer 800 by 600 by 256 colors - a much better choice for GIS related work.

Note that GSS Graphics Kernel Software (GKS) drivers must be available for any video board you attempt to use with MAYA and PC ARC/INFO; the highest GKS resolution for Super VGA offered now is 800x600x16 color (HIRES.SYS). The more expensive IBM 8514A (1024 by 768 by 256 colors) is another choice, for those that can afford it; note that this "standard" has now been replaced by the IBM XGA definition, which is also 1024 by 768 by 256 colors, interlaced.

Appendix I: MAYA's Hierarchical Menu Structure

DATA

DOS DIRECTORY LIST

CONVERT DATA SERVICES
- MAYA DBF To INFO Table
- INFO Table To MAYA DBF
- Raster Grid to DBF or Text

LIST COVERAGES

ARCHIVE SERVICES
- IMPORT a PC ARC/INFO Coverage
- EXPORT a PC ARC/INFO Coverage
- Archive (Squeeze) A Set of Files
- Restore Set of Archived Files
- View Directory of File Set Within Archive
- Backup Files To Backup Media

TELECOMMUNICATIONS SERVICES

MAINTAIN DATA TABLES
- View DBF Table Structure
- Append ASCII Data To Database Table
- Replace Data Table Field Values
- Delete Data Records Selectively
- Modify A Data Table Structure
- Erase A DBF, INFO Table or Vector Coverage

EDIT-VIEW

DBF TABLE EDIT-VIEW
- Select
- View
- Edit
- Create
  - Select A Pre-Defined Table Name
  - Introduce An Existing Attribute Table
  - Create A User-Defined Table Type
  - Create New Table Now

TEXT FILE EDIT-VIEW
- Select
- View
- Edit
- Create

GIS SERVICES

PC ARC/INFO COMMAND ACCESS

DESCRIBE COVERAGE
- Type of Coverage
- Enter Coverage Name
- Report Destination
- Begin Description Now

RASTERIZATION SERVICES
- Select An Existing Raster Map Definition
- Create New Raster Map Definition
- Edit Selected Raster Map Definition
- Delete Selected Raster Map Definition
- Begin Rasterization Now

```
                    GENERATE COVERAGE
                            New Coverage Name
                            Command File for Generate Job
                            Run Generate Now

                    QUICK PLOT VIEW

ANALYSIS SERVICES   MAYA RASTER ANALYSIS FUNCTIONS
                            MAYA Raster Analysis Startup Services
                                    Choose An Analysis Input Parameter Set
                                    View-Edit Selected Analysis Input Parameter Set
                                    Delete Selected Analysis Input Parameter Set
                                    Run Selected Analysis Now

                            Identify Point-In-Polygon Analysis
                            Point-To-Edge Distance Analysis
                            Sample Frame Analysis
                            Roving Window Simulation Analysis

                    ANALYSIS BATCH JOB SERVICES
                            Enable/Disable Analysis Input Sets
                            Delete All Disabled Analysis Input Sets
                            Run ALL Enabled Analysis Input Sets

                    DOS BATCH JOB SERVICES
                            Select An Existing Batch Job
                            Create A New Batch Job
                            Empty Out All Batch Jobs

                    RUN A USER-ASSIGNED PROGRAM

REPORTS             DATABASE TABLE PRINT

                    TEXT FILE PRINT

                    MOST RECENT REPORT

                    BEGIN PRINTING REPORT

SYSTEM              LOG SYSTEM ACCESS

                    DATA DICTIONARY SERVICES
                            View (Passively) Data Dictionary
                            Add-Edit Data Dictionary
                            Produce A Dictionary Report
                            Restore Dictionary Integrity

                    EDIT USER PROGRAM LIST

                    CONFIGURE DEFAULTS

                    SYSTEM STATUS

                    RUN ANY DOS COMMAND

QUIT                Enter (Q) Quit To DOS or (C) Cancel And Return to MAYA
```